PIC BASED MASTER-SLAVE MICROCONTROLLER BOARD DEVELOPMENT

JOE TOUK YONG

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

PIC BASED MASTER-SLAVE MICROCONTROLLER BOARD DEVELOPMENT

By

HOE FOOK YONG

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

PIC BASED MASTER-SLAVE MICROCONTROLLER BOARD DEVELOPMENT

by

Hoe Fook Yong

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Dr Irraivan A/L Elamvazuthi
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June, 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

(HOE FOOK YONG @ HSI FOOK YONG)

ABSTRACT

This report is about PIC Based Master-Slave Microcontroller Board Development. Currently, PS40B is used widely for controlling autonomous robots. PS40B is an enhanced version of PS40A. It is designed to control manual machine using Sony PS2 joystick. However, the PS40B using the 16F877A PIC microcontroller has its limitation of 40 pins and thus restricted to limited I/O. The research is to enhance the PIC16F877A and utilize the microcontroller by using Master & Slave concept and offer more I/O.

ACKNOWLEDGEMENTS

I would like to express my gratitude and thanks to my supervisor of this project, Assc. Prof Dr Irraivan Elamvazuthi for his assistance and guidance towards this research. He has constantly provided me with valuable guidance and advice which lead to the completion of this project.

Besides that, I also wanted to thank my co-supervisor, Mr Patrick Sebastian, for guiding me when I was wrong and I am able to learn from wrong to right.

Finally, I would like to thank University Technology Petronas, for providing me the platform to complete this project, with the first-class facilities, and superb environment to complete this project successfully.

Finally, I would like to apologize if any party was inadvertently excluded from being mentioned above and would like to thank all parties that were involved in making this project a success.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

PIC board is the platform for the PIC microcontroller to function. The PIC16F877A microcontroller is used for the board. In order to increase the effectiveness and reliability of the microcontroller, the PIC board is being designed according the application. For this project, the PIC board is designed to comply with the Petrobots application, which is the university Technology Petronas robotics organization. The board will be able to support and easily amend for any changes in future. Currently, PS40B is used widely for controlling autonomous robots. PS40B is an enhanced version of PS40A. It is designed to control manual machine using Sony PS2 joystick. However, the PS40B using the 16F877A PIC microcontroller has its limitation of 40 pins and thus restricted to limited I/O. This project will be able to enhance the PIC16F877A and utilize the microcontroller by using Master & Slave concept and offer more I/O[7].

1.1 Background of Study

To develop an autonomous robot, a microcontroller is needed as its "brain" in order for it to operate. For this project, PIC16F877A microcontroller is being used. University Technology Petronas used to buy all-ready board from vendors which have lots of limitation. The AR40B board, produced by Cyctron has been commonly used for robotics application and for the competition held every year. However, the AR40B has too many limitations[7]. Below described the limitations[3]:

- The AR40B using PIC16F877A only has 40 pins and offers 8 digital inputs and 3 analog inputs
- 8 units of relay with maximum current of 10 Amp to control 4 brush motors
- bi-directional.
- 2 designated I/O ports for driving 2 units of Oriental AXH series or

LINIX series of DC brushless motor(speed, encoder and direction)[8]

- 2 designated I/O ports for driving ordinary DC brush motor (speed and direction)
- 2 extra general purpose output pins
- 3 input switches for manual input
- The board was built as all-in-one. It will be difficult for maintenance and configuration
- High cost

Thus, by implementing this project, we can amend the board as our demand and applications.

1.2 Objectives

This project is to develop a controller board that can use two or more PIC16F877A microcontroller which offers more I/O by implementing Master Slave concept. Besides, the objective is also to develop a controller board based on UTP Petrobots applications. By doing so, this project can reduce the cost in purchasing controller board, easy to configure and maintenance

1.3 Scope of Study

This project uses the concept of Master & Slave. Two PIC16F877A microcontrollers will be used for this project in order to meet the objectives of have more I/Os.

CHAPTER 2
LITERATURE REVIEW

2.1 PIC16F877A[1]

PIC16F877A is a small piece of semiconductor integrated circuits. The package type of this integrated circuits is DIP package. DIP stand for Dual Inline Package for semiconductor IC. This package is very easy to be soldered onto the strip board. However using a DIP socket is much easier so that this chip can be plugged and removed from the development board.

PIC16F877A is very cheap. Apart from that it is also very easy to be assembled. Additional components that needed to make this IC work is just a 5V power supply adapter, a 20MHz crystal oscillator and 2 units of 22pF capacitors.

This IC can be reprogrammed and erased up to 10,000 times. Therefore it is very good for new product development phase.[1]

To start programming the PIC16F877A, firstly assemble the PIC16F877A circuit with the 5V power supply, 20MHz crystal oscillator and two 22pF capacitors. 5V voltage regulator circuit will be needed only have 12 volt power supply adapter is used. 5V voltage regulator circuit can be build using 7805 IC, IN4007 diode and two 10uF capacitors.

2.2 Master & Slave

This project is to enhance the PIC16F877A, by using the Master Slave Concept. Master & Slave concept is divided into two types which is I2C and SPI(Serial Peripheral Interface). However, for this project, SPI is being used due to its serial communication between two or more devices at high speed and is reasonably easy to implement.

2.2.1 Serial Peripheral Interface

SPI is a Synchronous protocol. The clock signal is provided by the master to provide synchronization. The clock signal controls when data can change and when it is valid for reading. Since SPI is synchronous, it has a clock pulse along with the data.

Since SPI has a clock signal, the clock can vary without disrupting the data. The data rate will simply change along with the changes in the clock rate. This makes SPI ideal when the microcontroller is being clocked imprecisely, such as by a RC oscillator.

SPI is a Master-Slave protocol and only the master device can control the clock line, SCK. No data will be transferred unless the clock is manipulated. All slaves are controlled by the clock which is manipulated by the master device. However, the slaves may not manipulate the clock.[2]

2.2.2 SPI Communicationi[2]

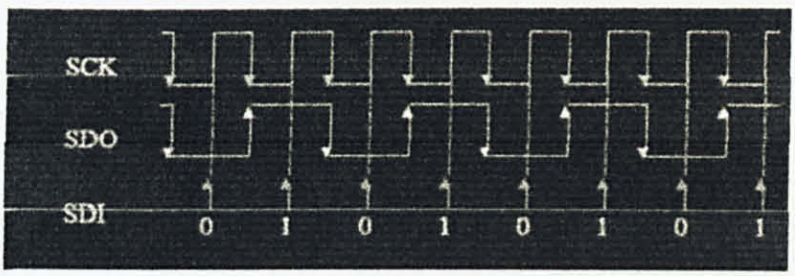


Figure 1 SPI communication

Figure 1 shows an example of SPI communication. The arrows indicate whether the signal is experiencing a rising or falling edge at the time. SDI shows

when the data is sampled. As can be clearly seen, the sampling is done on the opposite clock edge of when the data changes.

SPI is a Serial Interface and uses the following signals to serially exchange data with another device:

SS - This signal is known as Slave Select. When it goes low, the slave device will listen for SPI clock and data signals.

SCK - This is the serial clock signal. It is generated by the master device and controls when data is sent and when it is read

SDO - This is the Serial Data Output signal. SDO carries data out of a device.

SDI - SDI is the Serial Data Input line. It carries data into a device

2.3 PIC Pin Configuration

2.3.1 Master Pin Configuration

Figure 2 show the master pin configuration.

RESET BUTTON	1	MCLR	RB7/PGD	40	D/I
A/I	2	RA0/AN0	RB6/PGC	39	D/I
A/I	3	RA1/AN1	RB5	38	D/I
CONNECT TO SS	4	RA2/AN2	RB4	37	D/I
A/I	5	RA3/AN3	RB3/PGM	36	D/I
ENCODER 1	6	RA4	RB2	35	D/I
A/I	7	RA5/AN4	RB1	34	D/I
A/I	8	RE0/AN5	RB0	33	D/I
A/I	9	RE1/AN6	VDD	32	5V
A/I	10	RE2/AN7	VSS	31	GROUND
5V	11	VDD	RD7/PSP7	30	RELAY
GROUND	12	VSS	RD6/PSP6	29	RELAY
CRYSTAL OSCILLATOR	13	OSC1	RD5/PSP5	28	RELAY
CRYSTAL OSCILLATOR	14	OSC2	RD4/PSP4	27	RELAY
ENCODER 1	15	RC0/T10S0	RC7/RX/DT	26	PROGRAMMER
PWM2	16	RC1/TI0S1	RC6/CK/TX	25	PROGRAMMER
PWM1	17	RC2/CCP1	RC5/SDO	24	MASTER/SLAVE
MASTER/SLAVE	18	RC3/SCK/SCL	RC4/SDA	23	MASTER/SLAVE
RELAY	19	RD0/PSP0	RD3/PSP3	22	RELAY
RELAY	20	RD1/PSP1	RD2/PSP2	21	RELAY

Figure 2 Master Pin Configuration

Basically, the master pin configuration is made to comply with the objectives of the Petrobots, depending on application needed. From the Figure 2, it show the nature of the pin and the application of the pin is shown in the Table 4.

2.3.2 Slave Pin Configuration

Figure 3 show the master pin configuration.

RESET BUTTON	1	MCLR	RB7/PGD	40	D/I
A/I	2	RA0/AN0	RB6/PGC	39	D/I
A/I	3	RA1/AN1	RB5	38	D/I
CONNECT TO SS	4	RA2/AN2	RB4	37	D/I
A/I	5	RA3/AN3	RB3/PGM	36	D/I
ENCODER 1	6	RA4	RB2	35	D/I
A/I	7	RA5/AN4	RB1	34	D/I
A/I	8	RE0/AN5	RB0	33	D/I
A/I	9	RE1/AN6	VDD	32	5V
A/I	10	RE2/AN7	VSS	31	GROUND
5V	11	VDD	RD7/PSP7	30	RELAY
GROUND	12	VSS	RD6/PSP6	29	RELAY
CRYSTAL OSCILLATOR	13	OSC1	RD5/PSP5	28	RELAY
CRYSTAL OSCILLATOR	14	OSC2	RD4/PSP4	27	RELAY
ENCODER 1	15	RC0/T10S0	RC7/RX/DT	26	PROGRAMMER
PWM2	16	RC1/TI0S1	RC6/CK/TX	25	PROGRAMMER
PWM1	17	RC2/CCP1	RC5/SDO	24	MASTER/SLAVE
MASTER/SLAVE	18	RC3/SCK/SCL	RC4/SDA	23	MASTER/SLAVE
RELAY	19	RD0/PSP0	RD3/PSP3	22	RELAY
RELAY	20	RD1/PSP1	RD2/PSP2	21	RELAY

Figure 3 Slave Pin Configuration

Basically, the slave pin configuration is made to comply with the objectives of the Petrobots, depending on application needed. From the Figure 2, it show the nature of the pin and the application of the pin is shown in the Table 4.

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

3.1.1 Project Flow Chart

Figure 4 show the overall project flow chart.

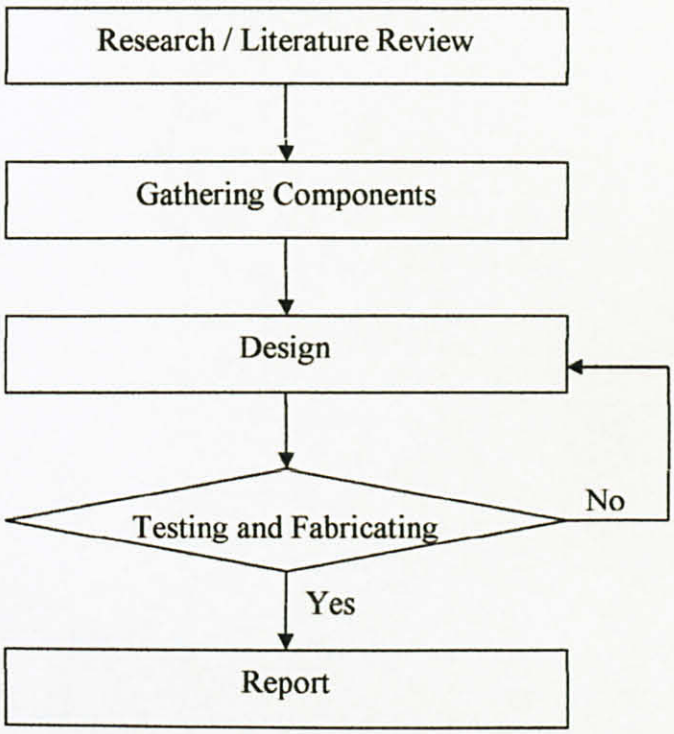


Figure 4 Project Flow Chart

Refer to Appendix A for this project's Gant Chart.

3.1.2 Gathering Information

First of all, information was gathered about the controller board and the concept of master & Slave. Basically the studies done by :

I. Research in the Internet
II. Find information from UTP Information Resource Center
III. Consulting the Lecturers

3.1.3 Gathering Components

All the components that needed for this project being identified and purchased.

3.1.4 Design

After research being made, a design was made and prototype was done on bread-board to test the board.

The microcontroller components is being designed as shown in Figure 5.

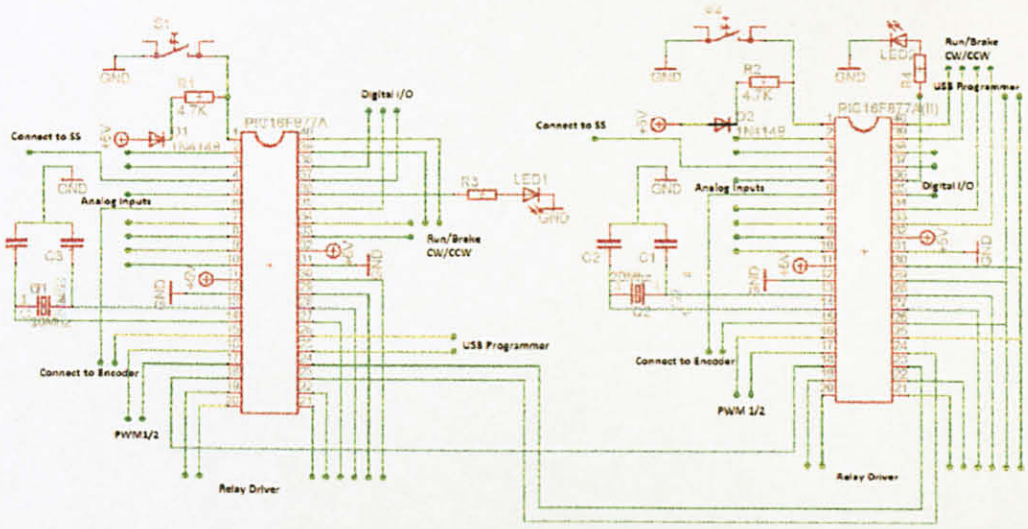


Figure 5 Schematic Diagram for Main Board

A clearer diagram can be referred at Appendix C.

Figure 6 show the schematic diagram for Power Supply Board.

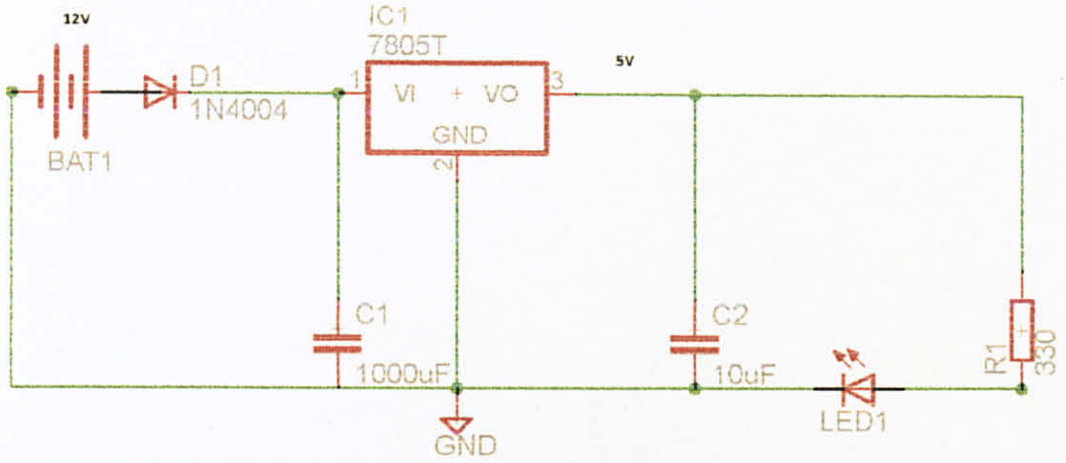


Figure 6 Schematic Diagram for Power Supply Board

This circuit was designed to produce three values of voltage which are 24V, 12V and 5V. 24V is produce from the sum of two batteries which are connected in series. 12V is produce from the voltage of a 12V battery while 5V is produce after it goes through the voltage regulator. The voltage regulator that would be used is LM 7805. The function of voltage regulator is to regulate the given voltage (12V) to 5V. Both capacitors, C1 and C2 are used to stabilize the voltage input and output of the PIC 16F877A. Power supply board would be connected to microcontroller board through connector.

3.1.5 Testing and Fabricating

The PIC16F877a is being programmed for both Master and Slave microcontroller. LED is being use as output and switches are being used as input. If the switch from Master Microcontroller Board, is press "ON", the LED, from Salve Microcontroller Board will lit. This shows that the signal is being transmit from Master Board and received at Slave Board.

The programming for Master Microcontroller is being shown in Appendix B
The programming for Master Microcontroller is being shown in Appendix C.

Table 1 shows the functions used for SPI programming.

Table 1 Functions used for SPI programming

Instruction	Description	Example
setup_spi (*modes*)	Initializes spi port	setup_spi (spi_master | spi_1_to_h | spi_clk_div_4) ;
spi_data_is_in ( )	Returns TRUE if data is received at the SPI port.	if (spi_data_is_in ( ))
spi_read (<char c>)	Returns a value read from the SPI port.	d = spi_read ( ) ;
Spi_write (<char c>)	Sets value of data to be written to SPI port	spi_write (d) ;

3.2 Tools and Software

Table 2 shows the components and software used:

Table 2 Software and Components used

<table>
<tr><th colspan="2">Components</th></tr>
<tr><td colspan="2">• Resistors
• Capacitors
• Diodes
• Connectors
• Crystal oscillator
• Transistor
• Voltage regulator
• Relays
• Fuses
• LEDs
• Switches
• Programmable Integrated Circuits (PICs)
• Encoders</td></tr>
<tr><th>Equipments</th><th>Software</th></tr>
<tr><td>• Power supply – batteries
• Brushless motor
• Brush motor
• Motor driver
• Sensors</td><td>• Eagle software
• Printed Circuit Board (PCB) Computer Aided Design (CAD) software
• Proteus</td></tr>
</table>

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Results

4.1.1 Power supply

For both PIC16F877A, same power supply is being used. The purpose for main power supply is to provide power for the circuit, relays, sensors and signal for brush motor. Brush motor use 1 x 12 volt battery as main power supply while brushless motor use 2 x 12 volt batteries. The batteries can be connected to the connectors JP1 and JP2. The function of the fuse is to protect short circuit between the two batteries. The main power supply can be switch on or off by toggle switch. The schematic diagram is show in Appendix B and Figure 9 show the implementation diagram of Power Supply Board.

Figure 7 Board for Power Supply

Table 3 Pins Number for J1 and J2 with the Connections

Pin Number	Symbol	Connection
1	+	Terminal Positive (+ve) of battery
2	-	Terminal negative (-ve) of battery

This circuit was designed to produce three values of voltage which are 24V, 12V and 5V. 24V is produce from the sum of two batteries which are connected in series. 12V is produce from the voltage of a 12V battery while 5V is produce after it goes through the voltage regulator. The voltage regulator that would be used is LM 7805. The function of voltage regulator is to regulate the given voltage (12V) to 5V. Both capacitors, C1 and C2 are used to stabilize the voltage input and output of the PIC 16F877A. Power supply board would be connected to microcontroller board through connector.

4.1.2 Microcontrollers

Microcontroller is where the programming of the autonomous robot will be downloaded. PIC 16F877A is being used with the implementation of Master & Slave using SPI mode. The schematic diagram is show in Appendix C and Figure 10 show the implementation for Microcontrollers Board.

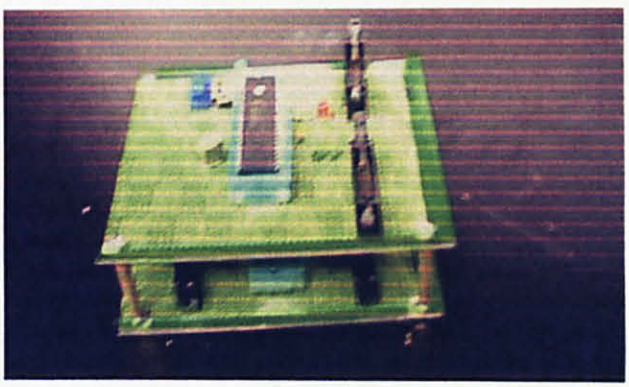


Figure 8 Microcontrollers Board

Table 4 List of PIC16F877A Pins and Functions for both Master and Slave Board

PIC Pins	PIC I/O	Functions
1	MCLR / VPP	Connect to reset button
11	VDD	Connect to 5V
32		
12	VSS	Connect to ground
31		
13	OSC1 / CLK1	Connect to crystal oscillator
14	OSC2 / CLK2	
4	SS	Connect to SS for SPI
2	RA0 / AN0	Connect to analog inputs
3	RA1 / AN1	
5	RA3 / AN3	
7	RA5 / AN4	
8	RE0 / AN5	
9	RE1 / AN6	
10	RE2 / AN7	
15	RCO / T1CKI	Connect to encoder 1
6	RA4 / T0CKI	Connect to encoder 2
18	RC3/SCK/SCL	Connect for SPI
23	RC4/SDA	Connect for SPI
24	RC5/SDO	Connect for SPI
35	RB2	Connect to digital inputs
37	RB4	
38	RB5	

36	RB3	Connect to LED I
19	RD0	Connect to relay driver
20	RD1	
21	RD2	
22	RD3	
27	RD4	
28	RD5	
29	RD6	
30	RD7	
34	RB1	Connect to RUN/BRAKE 1
33	RBO	Connect to CW/CCW 1
39	RB7	Connect to RUN/BRAKE 2
40	RB6	Connect to CW/CCW 1
16	RC1 / CCP2	Connect to PWM 2
17	RC2 / CCP1	Connect to PWM 1
25	RC6	Connect to USB programmer
26	RC7	

The function of reset button is to reset the microcontroller which is PIC16F877A. LED I would be connecting at pin 25, function as user program indicator LED. Crystal oscillator is used as inductive reactance where it creates an electrical signal with a very precise frequency. It would be connecting at pin 13 and pin 14. 20 MHz crystal oscillator was chose because to have maximum processing speed where PIC16F877A operating frequency is 20 MHz clock input[5]. Both capacitors are used to adjust the frequency at which will oscillate to avoid resonance frequency become higher[4].

4.1.3 Inputs

There will be four major inputs in this controller board. They are sensors, encoders, analog inputs and extended equipments. Figure 11 show the implementation of Input's board.

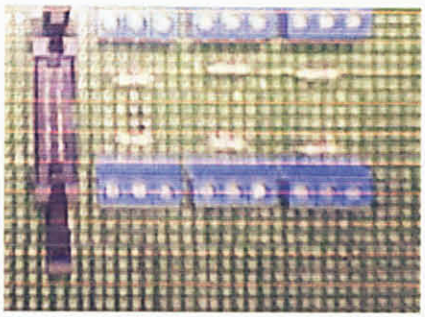


Figure 9 Board for Input

4.1.4 Outputs

There will be two major outputs in this controller board. They are brushless motor and brush motor[5][7]. Figure 12 show the implementation of output's board.

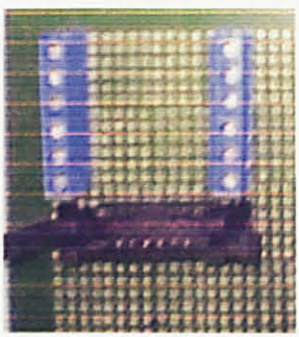


Figure 10 Board for Output

4.2 Programming

Figure 13 show the Coding for Master Microcontroller.

```
#include <16F877A.h>//
#fuses   NOWDT,NOPROTECT,XT,NOLVP,PUT
#use delay(clock=4000000)
float val=0;
int8 data=0;



setup_spi( SPI_MASTER | SPI_H_TO_L | SPI_XMIT_L_TO_H | SPI_CLK_DIV_16 );

while(1)
{

spi_write(data);
printf("\n\rdata = %u dataf = %f",data,val);

}
}
```

Figure 11 Coding for Master Microcontroller

The coding simply show how the master transmit the data by using the function "spi_write".

Figure 14 show the Coding for Slave Microcontroller.

```
//SLAVE program
#include <16F877A.h>//
#fuses NOWDT,NOPROTECT,XT,NOLVP,PUT
#use delay(clock=4000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)
int8 data=0;
void main()
{
setup_spi(SPI_SLAVE|SPI_H_TO_L|SPI_SS_DISABLED);
   output_high(PIN_C4);
   output_high (PIN_C5);
while(1)
      {
      if(SPI_DATA_IS_IN())
{
            data = spi_read();
            output_b(data):
            output_toggle(PIN_D4);
            printf("\n\r data = %u",data);
}
}
}
```

Figure 12 Coding for Slave Microcontroller

The coding simply show how the slave microcontroller read the data by using the function "spi_read". The sample of LED testing was shown in the Appendix

Both coding is important for SPI configuration to run. By having these coding, the Master will be able to transmit and slave will be able to receive.

4.3 Cost

Table 5 shows the total cost of components for the controller board:

Table 5 Total Cost of Components for the Controller Board

Board No.	Board	Quantity	Price per unit (RM)	Price (RM)
1	Power supply board	1	19.70	19.70
2	Microcontroller board	2	106.90	213.80
3	Analog inputs board	1	10.80	10.80
4	Digital inputs board	1	11.20	11.20
5	USB programmer board	1	43.70	43.70
			Total	**299.20**

Details of cost of specific board are shown in Appendix F, G, H, I, and J respectively.

4.4 Discussion

For this Project to be done, the main board, which is the Microcontrollers board has to be configure carefully. Programming has to be done to perfect the functionability of the board.

Microcontrollers Board for both Master and Slave was built by using stacking up concept. With this way, we can save space for large quantity of boards since SPI configuration is being used for this project. Besides that, the power supply is designed to produce three values of voltage which are 24V, 12V and 5V respectively. Here, we can see how all the boards are very competitive and can be used for many functions.

In this project, the I/O boards are designed in a very simple way, for inputs such as switches, outputs such as motors and etc. however, it is not restricted to these boards and other designed can be use too, with this Microcontroller Board.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In order to design a board, there are many steps need to be done accordingly. First of all, identify the suitable microcontroller and its availability. Then, we have to understand the nature of the microcontroller, how many I/Os it offers and lastly, designing part according to our application. Besides that, we also have to look into the programming carefully.

In the end, this project will be able to utilize the board to its maximum usage and able to achieve the objective of this project by offering more I/Os, user friendly and cheaper compared to AR40B.

5.2 Recommendation

For recommendation, the controller board can be constructed into printed circuit board (PCB) design to have more reliable final products. By soldering there might be some faulty.

REFERENCES

[1] Microchip Technology Inc., 2003, *PIC16&87XA, 28/40/44-pin Enhanced Flash microcontrollers*, U.S.A

[2] Lecture Note SPI

[3] Cyctron Technologies., Nov 2007, *Cytron AR40B Autonomous Robot Control Board Instruction Manual V1.3*, Johor, Malaysia

[4] Website refer to <http://en.wikipedia.org/wiki/Crystal_oscillator>

[5] Microchip Technology Inc., 2003, *PIC18F2455/2550/4455/4550, 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology*, U.S.A

[6] Cytron Technologies Sdn. Bhd., August 2008, *LINIX DC Brushless Motor User's Manual V1.5*, Johor, Malaysia

[7] Cytron Technologies Sdn. Bhd., January 2008, *Controlling DC Brush Motor using MD10A or MD30A V1.1*, Johor, Malaysia

APPENDICES

APPENDIX A

Gant Chart

	SEM 1														SEM 2													
Details / Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Selection of project title																												
Preliminary Research																												
Submission of Preliminary Report																												
Literature Review																												
Components Identification																												
Submission of Progress Report																												
Board Designing(Designing)																												
Seminar																												
Components Purchasing																												
Submission of Interim Report Final Draft																												
Oral Presentation																												
PIC Changed • PIC16F877A used • Master / Slave Concept																												
Research / Literature Review																												
Components Identification																												
Design Circuit and Board																												
Testing and Programming																												
Fabricating and Testing on Vero Board																												
Submission Progress Report																												
Poster Exibition																												
Draft Report Submission																												
Submission of Dissertion(Soft bound)																												
Final Presentation																												
Submission of Dissertion(Hard bound)																												

H1N1 BREAK

MIDSEM BREAK

MIDSEM BREAK

APPENDIX B

Example Coding for Master Microcontroller

```
/ This program demonstrates an SPI Master that can
// read bytes from a SPI slave. (Single bytes only).

#include <16F877.H>
#fuses XT, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, ERRORS)

#define SPI_SS   PIN_C2

// SPI mode definitions.
#define SPI_MODE_0 (SPI_L_TO_H | SPI_XMIT_L_TO_H)
#define SPI_MODE_1 (SPI_L_TO_H)
#define SPI_MODE_2 (SPI_H_TO_L)
#define SPI_MODE_3 (SPI_H_TO_L | SPI_XMIT_L_TO_H)


// SPI commands accepted by the SPI Slave PIC.
// The slave will respond to each of these commands
// with a single byte.  These must be the same
// values as defined in the Slave source code.
#define READ_ADC_CMD    1
#define READ_COUNTER_CMD 2
#define READ_SWITCH_CMD  3

//=====================================
void main()
{
char c;
int8 result;

output_high(SPI_SS);  // Initial Slave Select to a high level

// Initialize the hardware SSP for SPI Master mode.
setup_spi(SPI_MASTER | SPI_MODE_0 | SPI_CLK_DIV_4);

printf("Commands to slave are:\n\r");
printf("A: Read ADC\n\r");
printf("B: Read switch\n\r");
printf("C: Read counter\n\r");
printf("Press a key to send a command\n\r");

// If the user presses a key on the PC, get the
// character, convert it to an SPI command byte,
// and send it to the slave.  Then get the response
// byte from the slave and display it.
while(1)
  {
  c = getc();
  c = toupper(c);

  switch(c)
   {
    case 'A':    // Read ADC on Slave PIC
     output_low(SPI_SS);
     spi_write(READ_ADC_CMD);  // Send command to slave
     output_high(SPI_SS);

     delay_us(100);  // Give slave some time to respond

     output_low(SPI_SS);
     result = spi_read(0);    // Read response from slave
     output_high(SPI_SS);

     printf("ADC value = %X \n\r", result);
     break;

    case 'B':    // Read switch on Slave PIC
     output_low(SPI_SS);
```

```
    spi_write(READ_SWITCH_CMD); // Send command to slave
    output_high(SPI_SS);

    delay_us(100);

    output_low(SPI_SS);
    result = spi_read(0);    // Read response from slave
    output_high(SPI_SS);

    printf("Switch = %X \n\r", result);
    break;


  case 'C':    // Read counter on Slave PIC
    output_low(SPI_SS);
    spi_write(READ_COUNTER_CMD); // Send command to slave
    output_high(SPI_SS);

    delay_us(100);

    output_low(SPI_SS);
    result = spi_read(0);    // Read response from slave
    output_high(SPI_SS);

    printf("Counter = %X \n\r", result);
    break;


  default:
    printf("%c is an invalid command character.\n\r", c);
    break;
  }

} }
```

APPENDIX C

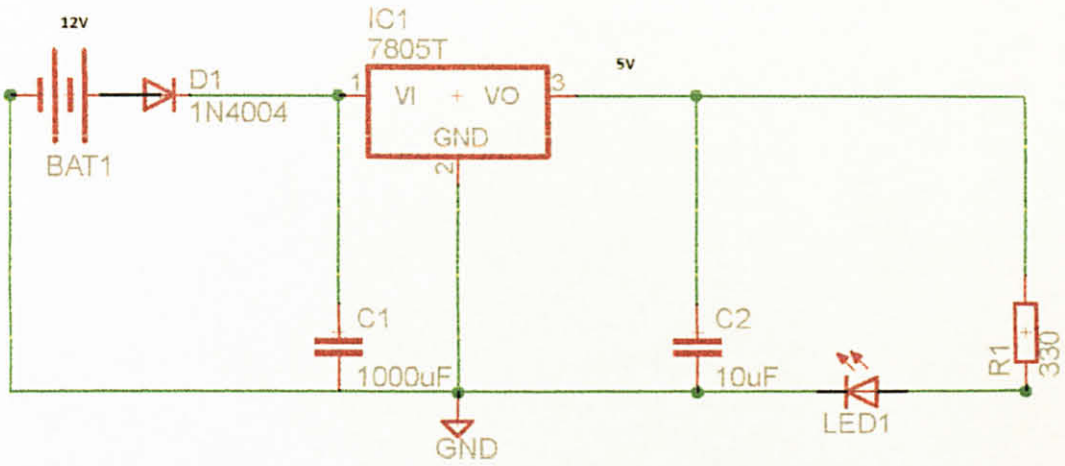
Coding for Slave Microcontroller

```
// This program demonstrates an SPI slave that can
// be read by the master.  Three different types
// of data (single bytes only) are returned upon
// command by the master.

#include <16F877.H>
#device adc=8
#fuses XT, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, ERRORS)

#byte SSPBUF = 0x13  // Register address for 16F877

// SPI modes for setup_spi() function.
#define SPI_MODE_0  (SPI_L_TO_H | SPI_XMIT_L_TO_H)
#define SPI_MODE_1  (SPI_L_TO_H)
#define SPI_MODE_2  (SPI_H_TO_L)
#define SPI_MODE_3  (SPI_H_TO_L | SPI_XMIT_L_TO_H)

// SPI commands accepted by this slave program.
// These must be non-zero values.
#define READ_ADC_CMD     1
#define READ_COUNTER_CMD 2
#define READ_SWITCH_CMD  3

// There is a push-button switch on pin B0, with a
// pull-up resistor on the pin.  Pressing the switch
// puts ground on the PIC pin.
#define SWITCH_PIN  PIN_B0

// This global variable is accessed both in the #int_ssp isr
// and in main().  It holds the current reading of the ADC.
int8 adc_result;

//-----------------------
// An int_ssp interrupt will occur whenever the SPI master
// transmits a byte to this slave PIC.  The following
// interrupt service routine (isr) decodes the command byte
// and sends back the appropriate response byte to the master.
// After sending the command byte with spi_write(), the master
// must call spi_read(0) to get the data byte.

#int_ssp
void ssp_isr(void)
{
int8 command;
static int8 counter = 0;
command = SSPBUF;

switch(command)  // Act on it
  {
   case READ_ADC_CMD:
    SSPBUF = adc_result;
    break;
   case READ_COUNTER_CMD:
    SSPBUF = counter;
    counter++;
    break;
   case READ_SWITCH_CMD:
    SSPBUF = input(SWITCH_PIN);
```

```
    break;
  } }



//=====================================
void main()
{
// Setup the ADC to read the voltage from the trimpot
// connected to pin A0.
setup_adc_ports(AN0);
setup_adc(ADC_CLOCK_DIV_8);
set_adc_channel(0);
delay_us(20);
adc_result = read_adc();
// Initialize the hardware SSP for SPI Slave mode 0.
setup_spi(SPI_SLAVE | SPI_MODE_0);
// Enable interrupts for the SPI slave.
clear_interrupt(INT_SSP);
enable_interrupts(INT_SSP);
enable_interrupts(GLOBAL);
// Update ADC value every 100 ms.
while(1)
  {
  adc_result = read_adc();
  delay_ms(100);
  }

}
```

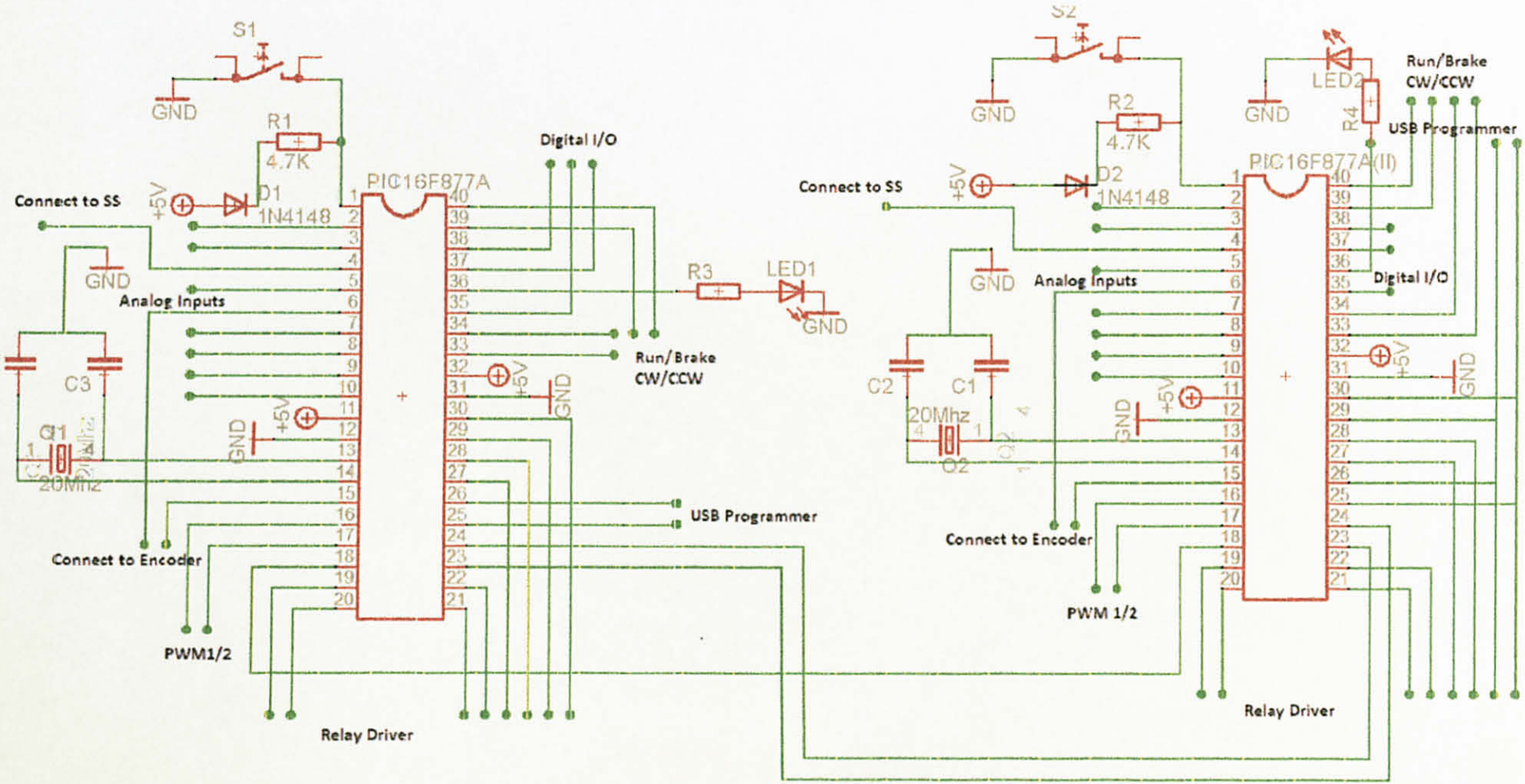
APPENDIX D

Schematic Diagram for Power Supply Board



APPENDIX E

Schematic Diagram for Microcontroller Board Master and Slave



APPENDIX F

Table 6 Costing for Power Supply Board

No.	Component	Quantity	Price per unit (RM)	Price (RM)
1	DG 128 – 2 ways	4	0.80	3.20
2	Diode 1N4007	2	0.30	0.60
3	Polarised capacitor 1000uF, 16V	2	0.70	1.40
4	Polarised capacitor 10uF, 16V	2	0.20	0.40
5	Resistor 330Ω	2	0.05	0.10
6	Toggle switch (SPDT)	2	1.00	2.00
7	Fuse (6Amp)	2	0.20	0.40
8	Fuse holder	2	0.60	1.20
9	LED (Green)	2	0.20	0.40
10	Voltage regulator LM7805	2	1.00	2.00
11	IDC socket – 10 pins	2	0.25	0.50
12	FRC plug – 10 pins	2	0.85	1.70
13	Ribbon wire (10 stripes)	50 cm	1.50	3.00
14	Vero board	1	2.50	2.50
			Total	**19.70**

APPENDIX G

Table 7 Costing for Components for Microcontroller Board

No.	Component	Quantity	Price per unit (RM)	Price (RM)
1	DG 128 – 2 ways	2	0.80	1.60
2	Diode 1N4148	2	0.25	0.50
3	Unpolarised capacitor 0.1uF	2	0.30	0.60
4	Unpolarised capacitor 30pF	4	0.30	2.40
5	Resistor 4.7kΩ	2	0.05	0.10
6	Resistor 330Ω	2	0.05	0.10
7	PIC16F877A – I/P	2	20.50	41.00
8	ZIF socket – 40 pins	2	17.00	34.00
9	Crystal Oscillator (20MHz)	2	1.30	2.60
10	Tact switch – 2 pins	2	0.50	1.00
11	LED (Green)	2	0.20	0.40
12	IDC socket – 10 pins	4	0.25	2.00
13	FRC plug – 10 pins	4	0.85	6.80
14	IDC socket – 14 pins	8	0.35	2.80
15	FRC plug – 14 pins	8	1.00	8.00
16	Donut board	2	1.50	3.00
			Total	**106.90**

APPENDIX H

Table 8 Cost of Components for Analog Inputs Board

No.	Component	Quantity	Price per unit (RM)	Price (RM)
1	DG 128 – 3 ways	6	0.60	3.60
2	PCB pin header (male)	1	0.50	0.50
3	Mini jumper (close)	1	0.30	0.30
4	IDC socket – 14 pins	1	0.40	0.40
5	FRC plug – 14 pins	1	1.00	1.00
6	Ribbon wire (25 stripes)	23 cm	2.50	2.50
7	Vero board	1	2.50	2.50
			Total	**10.80**

APPENDIX I

Table 9 Cost of Components for Digital Inputs Board

No.	Component	Quantity	Price per unit (RM)	Price (RM)
1	DG 128 – 3 ways	6	0.60	3.60
2	PCB pin header (male)	1	0.50	0.50
3	Mini jumper (close)	1	0.30	0.30
4	Resistor 4.7kΩ	8	0.05	0.40
5	IDC socket – 14 pins	1	0.40	0.40
6	FRC plug – 14 pins	1	1.00	1.00
7	Ribbon wire (25 stripes)	23 cm	2.50	2.50
8	Vero board	1	2.50	2.50
			Total	**11.20**

APPENDIX J

Table 10 Cost of Components for USB Programmer Board

No.	Component	Quantity	Price per unit (RM)	Price (RM)
1	DG 128 – 2 ways	1	0.40	0.40
2	DG 128 – 3 ways	1	0.60	0.60
3	USB socket	1	1.50	1.50
4	Diode 1N4148	4	0.25	1.00
5	Polarised capacitor 100uF, 16V	1	0.50	0.50
6	Polarised capacitor 47uF, 16V	1	0.40	0.40
7	Polarised capacitor 10uF, 16V	1	0.20	0.20
8	Polarised capacitor 1uF, 16V	2	0.15	0.30
9	Unpolarised capacitor 100nF	1	0.20	0.20
10	Unpolarised capacitor 15pF	2	0.15	0.30
11	Resistor 10kΩ	1	0.05	0.05
12	Resistor 4.7kΩ	2	0.05	0.10
13	Resistor 2.2kΩ	1	0.05	0.05
14	Resistor 1kΩ	1	0.05	0.05
15	Resistor 100kΩ	2	0.05	0.10
16	PIC 18F2550 – I/SP	1	27.00	27.00
17	IC socket – 28 pins	1	4.50	4.50
18	LED (Green)	1	0.20	0.20
19	Transistor BC548	2	0.50	1.00
20	Crystal Oscillator (12MHz)	1	1.30	1.30
21	IDC socket – 10 pins	1	0.25	0.25
22	FRC plug – 10 pins	1	0.85	0.85
23	Ribbon wire (10 stripes)	23 cm	1.50	0.35
24	Vero board	1	2.50	2.50
			Total	**43.70**